International Journal of Scientific Engineering and Technology
Volume No.4 Issue No.8, pp: 451-456

ISSN :2277-1581
01 August 2015

Performance Analysis of Weighted Encoding with Sparse Nonlocal Regularization and Spatially Adaptive Iterative Filtering Boost Denoising: A Review

Shivam Shukla* , Vijayshri Chaurasia**
ECE dept. of Maulana Azad National Institute of Technology Bhopal MP India
shivam1339@gmail.com*, vchaurasia@manit.ac.in**

Abstract: In this paper, we compare the spatially adaptive iterative filtering (SAIF) approach with Weighted Encoding with Sparse Nonlocal Regularization (WESNR) to maintain the denoising strength locally for any spatial domain method. These approaches has ability of filtering local image content iteratively using the given base filter, and the type of iteration and the iteration number are automatically optimized with respect to estimated risk (i.e., mean squared error). Experimental result shows that approx. 7% enhancement in the SNR of SAIF method as compared to the WESNR method.

Keywords: Image denoising, pixel aggregation, risk estimator, spatial domain filter.

I. Introduction

SINCE noise is a random phenomenon which occur basically all modern imaging systems in the process of transmission and reception, and denoising is used to restoration of fundamental image. There has been invented a lot number of denoising algorithms, and in general they are categories in two main parts: first one transform domain methods and second one spatial domain methods. Transform domain methods are used as a image is represented as a combination of certain transform basis function, by which signal to noise-ratio can be calculate and used the appropriate shrink for respective transform coefficient.

In Non-local mean (NLM) as we know it is data dependent filter, each pixel is calculated as the weighted average of all its similar pixels in the image, and by using similarity between them determined their weights. Zhang et al. [15] clustered the similar patches into a matrix and applied principal component analysis (PCA) to eliminate AWGN. The so-called LPG-PCA algorithm provides very good edge preservation performance. But in current years, In image restoration the sparse representation and dictionary learning based methods have been attracting significant attention. The combine use of sparse representation and nonlocal self-similarity regularization has led to idea of AWGN removal [26]. Another successful method called non-local means (NLM) extends the bilateral filter by replacing point-wise photometric distance with patch distances, which is more robust to noise [6]. In general, for spatial domain method the main problem is to determine the denoising strength. Because in this method denoising strength mainly depends on tuning parameter .A low value of controlling parameter would maintain high-frequency signal but will do minute denoising (estimation variance) in the image. And larger value of controlling parameter would quash more noise and also remove some important image information, and finally yield over-smoothed image. Another option for this controlling parameter selection is iterative filtering. By using this technique, either a filter estimated by wrong tuning parameter, but by using this filter method several numbers of times, can still get a well estimated output then iteration number then becomes another tuning parameter that needs to be carefully treated .In the spatial domain method, denoising strength is automatically adjusted according to local SNR. In [7] Milan far explains that a spatial domain denoising process can be equivalent as a transform domain filter, where the orthogonal basis elements are the eigenvectors of a symmetric and positive definite matrix determined by the filter; and the shrinkage coefficients are the corresponding eigenvalues ranging in [0, 1]. For NLM filters the eigenvectors corresponding to the dominant eigenvalues could well represent latent image contents. According to this idea, we suggest a spatially adapted iterative filtering (SAIF) strategy capable of controlling the denoising strength locally for any given spatial domain method. The proposed method iteratively filters local image patches, and the iteration method and iteration number are automatically optimized with respect to local MSE, which is estimated from the given image. To find out the MSE for each patch, we discuss plug-in risk estimator. While [9] also uses SURE to optimize the NLM kernel parameters, we illustrate that (1) the plug-in estimator can be superior for the same task, and (2) the adaptation approach can be extended to be spatially varying. This review paper is arranged as in Section II will provide some background, especially [7]'s proper analysis of spatial domain filters. Section III reviews two iterative methods to regulate the controlling (smoothing) strength for the filters in section III . Section IV describes SAIF strategy in detail. Experimental results are given in Section V to show the performance of the SAIF strategy using several leading filters. Finally we conclude this paper in Section VI.

II. Background

Assume the typical measurement model for the denoising difficulties: $y_i = z_i + e_i$ for $i = 1, \dots, n$ (1)

where $z_i = z(\boldsymbol{x}_i)$ is the positioned image at *position* $\boldsymbol{x}_i = [x_{i,}1, x_{i,}2]^T$, y_i is the noisy pixel value, and e_i denotes zeromean white noise with variance σ^2. The problem of denoising is to recover the set of underlying samples $\boldsymbol{z} = [z_1, \dots, z_n]^T$. The representation of vector model in the measurement model is:

$$y = z + e \qquad (2)$$

ISSN :2277-1581
01 August 2015

As explained in [5], [7] most spatial domain filters can be represented through the following non-parametric restoration framework:

$$\hat{z}_i = \arg\underbrace{min}_{z_i} \sum_{j=1}^{n} [z_i - y_i]^2 K(x_i, x_j, y_i, y_j) \quad (3)$$

where z_i stands the estimated pixel at position x_i, and the weight (or kernel) function K(·) measures the similarity between the samples y_i positioned at x_i and y_j positioned at x_j. In general the widely used kernel function is the Bilateral (BL) filter [4], that smooth images by the help of a nonlinear combination of nearby image values. The method combines pixel values based on both their geometric closeness and their photometric similarity. This kernel can be expressed in a separable fashion as follows:

$$K_{ij} = exp\left\{\frac{-\|x_i - x_j\|^2}{h_x^2} + \frac{-(y_i - y_j)^2}{h_y^2}\right\} \quad (4)$$

in which h_x and h_y are controlling (smoothing) parameters. The closely look like the bilateral filter except that the photometric similarity is collected in a patch wise manner:

$$K_{ij} = exp\left\{\frac{-\|x_i - x_j\|^2}{h_x^2} + \frac{-\|y_i - y_j\|^2}{h_y^2}\right\} \quad (5)$$

Where $\boldsymbol{y}_i$ patch positioned at y_i and $\boldsymbol{y}_j$ patch positioned at y_j. Only in theory (not in practical way,) the NLM kernel has the patch-wise photometric distance ($h_x \rightarrow \infty$).

In common way, preceding restoration algorithms are based on the same framework (3) in which some data-adaptive kernels are assigned to each pixel contributing to the filtering. Minimizing equation (3) gives a normalized weighted averaging process:

$$\hat{z}_i = \boldsymbol{w}_i^T \boldsymbol{y} \quad (7)$$

where the weight vector w_i is

$$w_i = \frac{1}{\sum_{j=1}^{n} K_{ij}} [K_{i1}, K_{i2}, \dots, K_{in}]^T \quad (8)$$

Collecting the weight vectors together, the filtering process is proceed for all the sample pixels simultaneously by a multiplication form of matrix-vector

$$\hat{\boldsymbol{z}} = \begin{bmatrix} w_1^T \\ w_2^T \\ \dots \\ w_1^T \end{bmatrix} y = \boldsymbol{W}\boldsymbol{y} \quad (9)$$

Where $\hat{\boldsymbol{z}}$ denote estimate signal and **W** denote the filter matrix. **W** is a positive row-stochastic matrix (sum of every row in matrix will be 1). This matrix will not be symmetric matrix but consist of positive real eigenvalue [7]. Even **W** is not a symmetric matrix in general, it can be closely estimated with a symmetric positive definite matrix [12]. The symmetrized **W** must also stay row-stochastic, that means we get a symmetric positive definite matrix which is doubly (i.e., row- and column-) stochastic. Thus we can estimate eigen –decomposition of symmetric W as follows:

$$\mathrm{W} = \mathrm{VSV}^{\mathrm{T}} \quad (10)$$

In this equation S = diag[$\lambda_1, \dots \dots, \lambda_n$] consist of the eigenvalues in descending order $0 \leq \lambda_1 \leq, \dots, < \lambda_n = 1$, and V is an orthogonal matrix V = [$v_1, \dots\dots, v_n$] consist of the corresponding eigenvectors of **W** in its columns. Because **V** is orthogonal, its columns specify a set of basis functions. therefore the filtering process can be expressed as:

$$\hat{\boldsymbol{z}} = \boldsymbol{W}\boldsymbol{y} = \boldsymbol{V}\boldsymbol{S}\boldsymbol{V}^T\boldsymbol{y} \quad (11)$$

Here by using the eigenvectors of **W** the input data **y** is convert into the domain spanned; then, each coefficient is modify (scaled) by the factor λ_i ; and an inverse transformation is done in final step, to get the output. By doing the above analysis we find that the denoising strength for every basis of a given filter is thus controlled by the scaled factor $\{\lambda_i\}$. In the next sections we will elaborate this concept more broadly.

III. Iterative Filtering Methods

Main shrinkage procedures based on various spatial domain filters have been discussed in [7], where statistical analysis shows that the optimal filter with respect to MSE is the local Wiener filter with $\lambda_i = \frac{1}{1+snr_i^{-1}}$, where snr_i indicate ith channel signal-to-noise ratio. In, the local Wiener filter require proper knowledge of the local signal to noise (SNR) of every basis channel, that does not directly accessible in general. In denoising schemes such as [1] and [3] the Wiener shrinkage criterion works centred on a pilot estimate of the latent image. Still, the Wiener filter's performance firmly believes on accuracy of this estimate. Iterative filtering can be a trustworthy substitute for reducing sensitivity of the basis shrinkage to the calculated local SNR. Then, the iteration number will be the only parameter to be locally optimized. To approach the locally optimal filter performance in a stable way, we propose the use of two iterative local operators; namely *diffusion* and *boosting*. In [13] we have shown that performance of any type of kernel could be enhanced by iterative diffusion which gradually removes the noise in each iteration. Yet, diffusion filtering also takes away latent details from the underlying signal. On the other hand, iterative boosting is a mechanism to preserve these lost details of the signal. By using the two iterative filtering methods, we can avoid either over-smoothing or under-smoothing due to incorrect parameter settings. In another way, these two methods deliver a way to start with any filter, and maintain the values of shrinkage factors $\{\lambda_i\}$ to find a better and firm approximation of the Wiener filter. Further we have explained the two methods, separately.

A. Diffusion

The notion of diffusion in image filtering was originally followed by the basic idea of heat propagation and explained using a partial differential equation. this method adopt the discrete version of it, that is easily represented by repeated applications of the same filter as described in [7]

$$\hat{\boldsymbol{z}}_k = \boldsymbol{W}\hat{\boldsymbol{z}}_{k-1} = \boldsymbol{W}^k\boldsymbol{y}. \quad (12)$$

Each application of ***W*** can be interpreted as one step of anisotropic diffusion with the filter ***W***.By taking large value of iteration number of *k* will eliminate the noise and high frequency component of noise, over smooth the image at the same time and small number of iteration *k* maintain the underlying structure. Minimization of MSE (or more accurately an estimate of it) stands the proper time to stop filtering, that will avoid becoming under- or over- smoothing of image.

Here ***W*** is symmetric, the filter in the iterative model (12) can be decomposed as:

$$\boldsymbol{W}^k = \boldsymbol{V}\boldsymbol{S}^k\boldsymbol{V}^T \quad (13)$$

ISSN :2277-1581
01 August 2015

Here S^k = diag$[\lambda_1^k, \ldots, \lambda_n^k]$. It is important that in spite of the general prediction of *k* as a discrete step, by the realization spectral decomposition of W^k makes it possible to substitute *k* with any positive real number. The hidden image **z** would be in the column space of **V** as $\mathbf{b} = \mathbf{V}^\mathrm{T}\mathbf{z}$, where $\mathbf{b} = [b1, b2, \ldots, bn]^\mathrm{T}$, and $\{b_i^2\}$ stands the projected signal energy over all the eigenvectors. As shown in [7] the iterative estimator $\hat{z}_k = W^k y$ has the following squared bias:

$$\|bias_k\|^2 = \|(I - W^k)z\|^2 = \sum_{i=1}^n (1-\lambda_i^k)^2 b_i^2 \qquad (14)$$

Correspondingly, the estimator's variance is:

$$\mathrm{var}(\hat{z}_k) = \mathrm{tr}(\mathrm{cov}(\hat{z}_k)) = \sigma^2 \sum_{i=1}^n \lambda_i^{2k} \qquad (15)$$

Finally, the MSE is given by

$$MSE_k = \|bias_k\|^2 + \mathrm{var}(\hat{z}) = \sum_{i=1}^n (1-\lambda_i^k)^2 b_i^2 + \sigma^2 \lambda_i^{2k} \qquad (16)$$

As the iteration number *k* grows, the bias term increases, but the variance decays to the constant value of σ^2. Of course, bthis expression for the MSE is not practically useful yet, since the coefficients $\{b_i^2\}$ are not known. Later we describe a way to estimate the MSE in practice. But first, let us introduce the second iterative mechanism which we will employ. Boosting is discussed in the following and as we will see, its behaviour is quite different from the diffusion filtering.

B. Boosting

Meanwhile diffusion filtering is used in more practice but in low SNR case diffusion filtering method is fail. And in that case, diffusion always eliminates some part of the noise and signal, concurrently. This problem is short out by boosting that *recycles* the removed components of signal from the residuals, in each iteration. Defining the residuals as the difference between the estimated signal and the noisy signal: $r_k = y - \hat{z}_{k-1}$, the iterated estimate can be expressed as

$$\widehat{z_k} = \hat{z}_{k-1} + W_{rk} = \sum_{j=0}^k W(I-W)^j y$$
$$= (I - (I-W)^{k+1})y \qquad (17)$$

where $\hat{z}_0$= **Wy**. And as *k* increases, the estimate backs to the noisy signal **y**. basically the boosting filter has different behavior than the diffusion filter where after each iteration the calculated signal gets close to a constant. After *k* iteration the squared magnitude of the bias is

$$\|bias_k\|^2 = \|(I-W)^{k+1}z\|^2 = \sum_{i=1}^n (1-\lambda_i)^{2k+2} b_i^2 \qquad (18)$$

Estimators' variance is,

$$\mathrm{Var}(\hat{z}_k) = \mathrm{tr}(\mathrm{cov}(\hat{z}_k)) = \sigma^2 \sum_{i=1}^n [1 - (1-\lambda_i)^{k+1}]^2 \qquad (19)$$

the overall MSE is

$$MSE_k = \sum_{i=1}^n (1-\lambda_i)^{2k+2} b_i^2 + \sigma^2 (1 - (1-\lambda_i)^{k+1})^2 \qquad (20)$$

As *k* increases, the variance will increase and bias term will decrease. Highlighting the nature of the diffusion iteration, analyse that *boosting replace diffusion when diffusion unable to improve the filtering performance*. The support of this task is that we can automatically optimize the type and number of iterations locally to boost the performance of a given base filtering method at the same time.

Algorithm of WESNR

Here, the dictionary φ is given, assume. It is an important issue selection of dictionary to the sparse coding and reconstruction of a signal. In specific, from natural image, learning dictionaries patches has given favourable results in image restoration [17, 18]. In this algorithm, the dictionary φ is pre-learned from clean natural images, and the pixels corrupted by Impulsive Noise (IN) will have big coding residuals. Consequently, the coding residual e_i can be used to guide the setting of weight W_{ii}, and W_{ii} should have inverse relation with the strength of e_i. In order to make the weighted encoding stable and easy to control, we select Wii should lie [0; 1]. So, relation between W_{ii} and e_i is

$$W_{ii} = exp(-a e_i^2); \qquad (21)$$

Here W_{ii} should be decreasing rate with respect to e_i.To maintain this relation ***a*** will be positive constant By Eq. (21), the pixels corrupted by IN will be adaptively assigned with lower weights to reduce their impact in the process of encoding. Let V be a diagonal matrix and initialize it first as an identity matrix, and then in the (k + 1)th iteration, each element of V is restructured as

$$V_{ii}^{k+1} = \lambda / [(\alpha_i^{(k)} - \mu_i)^2 + \varepsilon^2]^{1/2} \qquad (22)$$

where ε is a scalar and $\alpha_i^{(k)}$ is the i^{th} element of coding vector α in the k^{th} iteration. Then we update α as

$$\hat{\alpha}^{k+1} = (\varphi^T W \varphi + V^{(k+1)})^{-1} (\varphi^T W y - \varphi^T W \varphi \mu) + \mu \qquad (23)$$

By iteratively updating V and α, the desired α can be efficiently obtained. A number of patches (size: 7×7) are extracted from the five images and they are clustered into 200 clusters by using the K-means clustering algorithm. For each cluster, a compact local PCA dictionary is learned. Meanwhile, the centroid of each cluster is calculated. For a given image patch, the Euclidian distance between it and the centroid of each cluster is computed, and the PCA dictionary associated with its closest cluster is chosen to encode the given patch. Note that since the selected dictionary, denoted byφ_i, is orthogonal, the μ_i for patch x_i can be simply computed as $\mu_i, = \varphi_i^T \hat{x}_i$.

First the dictionary φ is adaptively calculated for a given patch, the WESNR model can be solved by iteratively updating W and α. The updating of W depends on the coding residual e. In the literature of mixed AWGN and SPIN noise removal [19]-[22], AMF [2] is widely used to detect SPIN. In order to make a fair comparison with them, in the case of AWGN+SPIN noise removal, we apply AMF to y to obtain an initialized image $x(0)$, and then initialize e as:

$$e^{(0)} = y - x^{(0)} \qquad (24)$$

In the case of AWGN+RVIN+SPIN noise removal, AMF cannot be applied to y to initialize x. We initialize e as

$$e^{(0)} = y - \mu_y.1 \qquad (25)$$

where μ_y is the mean value of all pixels in y and 1 is a column vector whose elements are all 1. In other words, we simply use the mean value of y to initialize x. Then the initial coding residual can be roughly computed. This simple initialization strategy works very well in all our experiments. With the initialized coding residual $e^{(0)}$, W can be initialized by Eq. (21). The main procedures of the WESNR based mixed noise removal algorithm are brief in Algorithm 1. In our algorithm, we set t = $\|\Phi\alpha^{(k+1)} - \Phi\alpha^{(k)}\| / \|\Phi\alpha^{(k)}\| < \tau$ as the termination condition.

The code for the WESNR algorithm as:

Algorithm 1: WESNR method to eliminate mixed noise

Input: Dictionaryφ, noisy image y;

Initialize e by Eq. (24) (or Eq. (25)) again

Initialization of μ to 0

Initialization of W by Eq. (21);

Output: Denoisey image x.
Loop: iterate on $k = 1, 2, \ldots, K$;
1. Find out $\alpha^{(k)}$ by Eq. (23);
2. Find out $\alpha^{(k)} = \varphi\alpha^{(k)}$ and accordingly change the nonlocal coding vector μ;
3. Find out the residual $e^{(k)} = y - x^{(k)}$;
4. Compute the weights value W by $e^{(k)}$ using Eq. (21);
End
Output will be denoisey image $x = \varphi\alpha^{(k)}$.

IV. Modified Saif Method

Based on the analysis from Section III we propose an image denoising strategy which, given any filter using the framework (3), can boost its performance by utilizing its spatially adapted transform and by employing an optimized iteration method. This iterative filtering is implemented patch wise, so that it is capable of automatically adjusting the local smoothing strength according to local SNR.

A. Optimal Iteration Estimation

Let us assume a patch **y** and corresponding filter matrix **W**, here main aim is to select appropriate iteration method and number of iteration that will minimize the MSE. In brief, the optimal stopping time $\hat{k}$ for each iteration method is given as:

$$\hat{k} = \underset{k}{\operatorname{argmin}}\ MSE_k \tag{21}$$

The method to calculate an unbiased estimate of MSE is the SURE [8].The best alternative, here we propound, is the plug-in risk estimator, that is biased and works based on an estimate of the local SNR. First, we estimate, eigenvalues and eigenvectors of the filter from a *pre-filtered* patch $\tilde{\mathbf{z}}$, obtained with the help of the base filter by setting some arbitrary parameter. We have expressed as:

$$\mathbf{W}(\tilde{\mathbf{z}}) = \mathbf{VSV^T} \tag{22}$$

In spite of the earlier prediction of k as a discrete step, in practical way the spectral decomposition of W^k indicate that, k can be any positive real number. To be more definite, $\mathbf{W}^k = \mathbf{VS}^k\mathbf{V}^T$, with $\mathbf{S}^k$ = diag[$\lambda_1^k, \ldots\ldots, \lambda_n^k$] where k is positive real number. In practical implementation, the filter can be suggested with modified eigenvalues for any $k > 0$. This will improve the performance .In consequence; a real-valued k automatically takes and smoothly adjusts the local bandwidth of the filter. While decreasing the iteration number k can be predicted as smaller tuning parameter h_y for NLM kernel, larger the value k is equivalent to a wider kernel. Further, we explain the two risk estimators and find out that the plug-in estimator will give better performance in comparison to the SURE estimator for estimated local SNR.

1) Plug-In Risk Estimator: It is explain in Algorithm 2. In this technique, risk estimators for boosting and diffusion are calculated based on the pre filtered patch $\tilde{\mathbf{z}}$, computed using the base filter with arbitrary parameters. More clearly, for the estimation of the signal coefficients as follow:

$$\hat{b} = V^T\tilde{z} \tag{23}$$

This equation's contribution can be predicted as equipping the risk estimator with some prior idea of the local SNR of the image. The estimated signal coefficients allow us to use (24) and (25) to find out MSE_k in each patch: Diffusion Plug-in Risk Estimator is given as:

$$Plug - in_k^d = \sum_{i=1}^{n}(1-\lambda_i^k)^2\hat{b}_i^2 + \sigma^2\lambda_i^{2k} \tag{24}$$

Boosting Plug-in Risk Estimator is given as:

$$Plug - in_k^b = \sum_{i=1}^{n}(1-\lambda_i)^{2k+2}\hat{b}_i^2 + \sigma^2(1-(1-\lambda_i)^{k+1})^2 \tag{25}$$

In each patch, minimum values of $Plug - in_k^d$ and $Plug - in_k^b$ as a function of k are calculated and then compared, and the iteration type with the least risk is selected. Since the optimal iteration number $\hat{k}$can be any real positive value, in the application of the diffusion filter, $\mathbf{W}^{\hat{k}}$ is substitute by $\mathbf{VS}^{\hat{k}}\mathbf{V}^T\mathbf{y}$ where $\mathbf{S}^{\hat{k}}$ = diag[$\lambda_1^{\hat{k}}, \ldots\ldots\ldots, \lambda_n^{\hat{k}}$]. This has been similarly shown for the boosting filter in Algorithm 1. Next, for the sake of comparison, the SURE estimator is discussed.

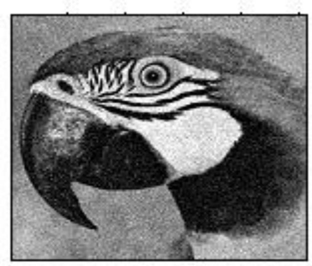
Algorithm 2: Plug-in Risk Estimator
Input: Noisy Patch: **y**, Pre-filtered Patch: $\tilde{z}$, Patch Filter: **W**
Output: Denoised Patch: $\hat{\mathbf{z}}$

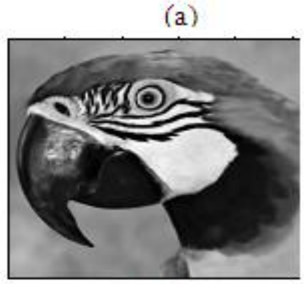
1. Eigen-decomposition of the filter $\mathbf{W}(\tilde{z}) = \mathbf{VSV^T}$;
2. **B** = $\mathbf{V^T}\tilde{z}$ ⇐ Compute the signal coefficients;
3. $Plug - in_k^d$, $Plug - in_k^b$ ⇐ Compute the estimated risks;
4. **if** min{$Plug - in_k^d$} < {$Plug - in_k^b$}
5. $\hat{k}$= arg $\underset{k}{min}$ $Plug - in_k^{df}$ ⇐ Diffusion optimal iteration number;
6. $\hat{\mathbf{z}} = \mathbf{VS}^{\hat{k}}\mathbf{V}^T\mathbf{y}$ ⇐ Diffusion patch denoising;
7. **else**
8. $\hat{k} = \underset{k}{\operatorname{argmin}}\ Plug - in_k^{bs}$ ⇐ Boosting optimal iteration number;
9. $\hat{\mathbf{z}} = \mathbf{V}\left(\mathbf{I} - (\mathbf{1} - \mathbf{S})^{\hat{k}+1}\right)\mathbf{V}^T\mathbf{y}$ ⇐ Boosting patch denoising;
10. **End**

V. Experimental Results

In this section we evaluate performance of method for denoising various images .we first compare results with ones from the standard kernels: NLM and Bilateral.


(a)


(b)


(c)


(d)

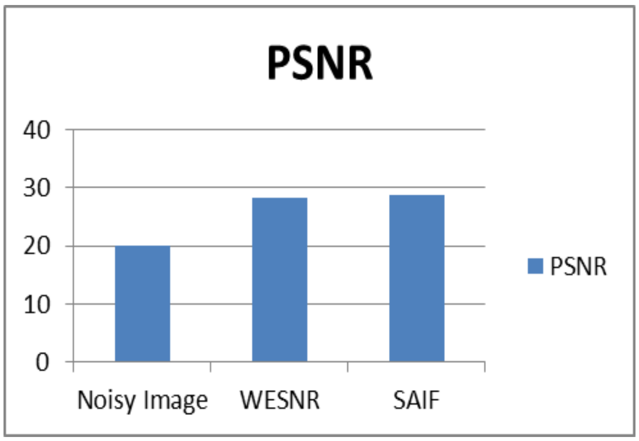
ISSN :2277-1581
01 August 2015

Fig.10. Comparison of denoising performance on noisy *Parrot* image corrupted by AWGN of $\sigma = 25$. (a) Original image. (b) Noisy input. (c) WESNR [6]. (d) SAIF (NLM).

TABLE-I
The PSNR Comparison between WESNR and SAIF Methods

Method	PSNR	Time (Sec.)
Noisy Image	20.15	---
WESNR	27.18	23.37
SAIF	28.87	90.22

We also test stability of SAIF when an arbitrary tuning parameter is used. In all cases the estimators show a promising improvement over the standard kernels. We will show that the resulting SAIF-ly improved filters are comparable, in terms of MSE (PSNR) and SSIM [14], to advanced denoising methods, and in many cases visually better.



As shown, the diffusion gets better most of the flat and smooth patches, boosting cares the texture and more complex ones. It is pointed that applying overlapped patches has the advantage of computing multiple estimates for each pixel from the *both* iteration types. In another way, in the aggregation process, some pixels may be from diffusion in one patch, whereas others may be from boosting in another overlapping patch. This is intentionally helpful for pixels at the border of smooth and texture regions.

V. Conclusion

For any spatial domain filter, we can boost its performance highly developed by employing optimized iteration methods. Patch-wise implementation is followed in iterative filtering. Possessed with diffusion and boosting as two complementary iteration techniques, the optimum local filter filtered each patch. More precisely, by using the best iteration number and method that minimizes MSE in each patch, SAIF is capable of automatically adjusting the local smoothing strength according to local SNR. The weighted encoding (WESNR) technique gets rid of the Gaussian noise and impulse noise at a time. But the final result in terms of PSNR the SAIF gives 7% better performance.

References

i. *J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," IEEE Trans. Image Process., vol. 12, no. 11, pp. 1338–1351, Nov. 2003.*

ii. *K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," IEEE Trans. Image Process., vol. 16, no. 8, pp. 2080–2095, Aug. 2007.*

iii. *D. D. Muresan and T. W. Parks, "Adaptive principal components and image denoising," in Proc. IEEE Int. Conf. Image Process., Sep. 2003, pp. 101–104.*

iv. *L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," Pattern Recognit., vol. 43, pp. 1531–1549, Apr. 2010.*

v. *M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," IEEE Trans. Image Process., vol. 15, no. 12, pp. 3736–3745, Dec. 2006.*

vi. *P. Chatterjee and P. Milanfar, "Clustering-based denoising with locally learned dictionaries," IEEE Trans. Image Process., vol. 18, no. 7, pp. 1438–1451, Jul. 2009.*

vii. *P. Chatterjee and P. Milanfar, "Is denoising dead?" IEEE Trans. Image Process., vol. 19, no. 4, pp. 895–911, Apr. 2010.*

viii. *C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in Proc. IEEE Int. Conf. Comput. Vis., Jan. 1998, pp. 836–846.*

ix. *H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," IEEE Trans. Image Process., vol. 16, no. 2, pp. 349–366, Feb. 2007.*

x. *A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," Multiscale Model. Simulat., Int. J., vol. 4, no. 2, pp. 490–530, 2005.*

xi. *C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patchbased image denoising," IEEE Trans. Image Process., vol. 15, no. 10, pp. 2866–2878, Oct. 2006.*

xii. *J. Boulanger, C. Kervrann, and P. Bouthemy, "Space-time adaptation for patch-based image sequence restoration," IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 6, pp. 1096–1102, Jun. 2007.*

xiii. *P. Milanfar, "A tour of modern image filtering," IEEE Signal Process. Mag., to be published.*

xiv. *S. Ramani, T. Blu, and M. Unser, "Monte-Carlo SURE: A blackbox optimization of regularization parameters for general denoising algorithms," IEEE Trans. Imag. Process., vol. 17, no. 9, pp. 1540–1554, Sep. 2008.*

xv. *X. Zhu and P. Milanfar, "Automatic parameter selection for denoising algorithms using a no-reference measure of image content," IEEE Trans. Imag. Process., vol. 19, no. 12, pp. 3116–3132, Dec. 2010.*

xvi. *C. Stein, "Estimation of the mean of a multivariate normal distribution," Ann. Statist., vol. 9, no. 6, pp. 1135–1151, Nov. 1981.*

xvii. *M. Van De Ville and D. Kocher, "Nonlocal means with dimensionality reduction and sure-based parameter selection," IEEE Trans. Imag. Process., vol. 20, no. 9, pp. 2683–2690, Sep. 2011.*

xviii. *E. Seneta, Non-Negative Matrices and Markov Chains. Berlin, Germany: Springer-Verlag, 1981.*

xix. *R. A. Horn and C. R. Johnson, Matrix Analysis. Cambridge, MA: Cambridge Univ. Press, 1991.*

xx. *P. Milanfar, "Symmetrizing smoothing filters," SIAM J. Imag. Sci., to be published.*

xxi. *H. Talebi and P. Milanfar, "Improving denoising filters by optimal diffusion," in Proc. ICIP, Appear Conf., 2012, pp. 1–4.*

ISSN :2277-1581
01 August 2015

xxii. *J. Salmon and E. Le Pennec, "NL-Means and aggregation procedures," in Proc. ICIP Conf., Nov. 2009, pp. 2977–2980.*

xxiii. *A. S. Dalalyan and A. B. Tsybakov, "Aggregation by exponential weighting, sharp pac-bayesian bounds and sparsity," Mach. Learn., vol. 72, nos. 1–2, pp. 39–61, 2008.*

xxiv. *Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Trans. Imag. Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.*

xxv. *L. Zhang, W. S. Dong, D. Zhang, and G. M. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," Pattern Recognit., vol. 43, no. 4, pp. 1531-1549, Apr. 2010.*

xxvi. *W. S. Dong, L. Zhang, G. M. Shi, and X. Li, "Nonlocally Centralized sparse representation for image restoration," IEEE Trans. Image Process., vol. 22, no. 4, pp. 1620-1630, Apr. 2013.*

xxvii. *M. Aharon, M. Elad, and A.M. Bruckstein, "K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation," IEEE Trans. Signal Processing., vol. 54, no. 11, pp. 4311-4322, Nov. 2006.*

xxviii. *W. S. Dong, L. Zhang, G. M. Shi, and X. L. Wu, "Image deblurring and super-resolution by adaptive sparse domain selecion and adaptive regularization," IEEE Trans. Image Process., vol. 20, no. 7, pp. 1838-1857, Jul. 2011.*

xxix. *J. F. Cai, R. Chan, and M. Nikolova, "Two-phase methods for deblurring images corrupted by impulse plus gaussian noise," Inverse Problem Imaging., vol. 2, no. 2, pp. 187-204. 2008.*

xxx. *J. F. Cai, R. Chan and M. Nikolova, "Fast two-phase image deblurring under impulse," Journal of Mathematical Image and Vision., vol. 36, no. 1, pp. 46-53, Jan. 2010.*

xxxi. *Y. Xiao, T. Y. Zeng, J. Yu and M. K. Ng, "Restoration of images corrupted by mixed Gaussian-impulse noise via l1-l0 minimization," Pattern Recognit., vol. 44, no. 8, pp. 1708-1720, Aug. 2010.*

xxxii. *P. Rodr´_guez, R. Rojas and B. Wohlberg, "Mixed Gaussian-impulse noise image restoration via total variation," in IEEE. Int. Conf. Acoustics. Speech. and Signal Processing, 2012, pp. 1077-1080.*

xxxiii. *H. Hwang and R. A. Haddad, "Adaptive median filters: New algorithm and results," IEEE Trans. Image Process., vol. 4, no. 4, pp. 499-502, Apr. 1995.*